UTM

UNIVERSITI TEKNOLOGI MALAYSIA

FACULTY OF COMPUTING
SEMESTER II, SESSION 2024/2025

BACHELOR OF COMPUTER SCIENCE (BIOINFORMATICS)

SECB3203 PROGRAMMING FOR BIOINFORMATICS - SECTION 01

PROJECT
ALZHEIMER'S DISEASE PREDICTION

GROUP MEMBERS	MATRIC NO
TAN ZHAO HONG	A23CS0188
CHIN PEI WEN	A23CS0065
KOO XUAN	A23CS0300

LECTURER'S NAME : DR. SEAH CHOON SEN

SUBMISSION DATE : 03 DECEMBER 2025

Table of Contents

3.0 Flowchart of the Proposed Approach

3.3 Model Evaluation

Model evaluation is conducted to measure the effectiveness and reliability of the trained machine learning models in predicting Alzheimer's disease.

The process of measuring a trained model's performance on unseen data using metrics like accuracy, precision, recall, F1-score, cross-validation scores, and ROC-AUC.

```
from sklearn.model_selection import train_test_split, cross_val_score, GridSearchCV
from sklearn.preprocessing import StandardScaler, LabelEncoder
from sklearn.metrics import (accuracy_score, precision_score, recall_score, f1_score, confusion_matrix, classification_report, roc_auc_score, roc_curve)
```

3.3.1 Evaluation Metrics

To evaluate the performance of the classification models, several evaluation metrics were used, including accuracy, precision, recall, and F1-score. Accuracy measures the overall correctness of the model, while precision and recall provide insight into false positive and false negative predictions. The F1-score combines precision and recall to give a balanced evaluation of model performance.

```
325        # Print results
326        print(f"Accuracy:  {accuracy:.4f}")
327        print(f"Precision: {precision:.4f}")
328        print(f"Recall:    {recall:.4f}")
329        print(f"F1-Score:  {f1:.4f}")
330
```

Figure : Evaluation metrics of classification models.

```
================================================================================
MODEL TRAINING AND EVALUATION
================================================================================

Logistic Regression:
----------------------------------------
Accuracy:  0.8140
Precision: 0.8145
Recall:    0.8140
F1-Score:  0.8142
CV Accuracy: 0.8395 (+/- 0.0211)

Decision Tree:
----------------------------------------
Accuracy:  0.8884
Precision: 0.8884
Recall:    0.8884
F1-Score:  0.8884
CV Accuracy: 0.9069 (+/- 0.0117)

SVM:
----------------------------------------
Accuracy:  0.8326
Precision: 0.8305
Recall:    0.8326
F1-Score:  0.8302
CV Accuracy: 0.8336 (+/- 0.0169)

KNN:
----------------------------------------
Accuracy:  0.7465
Precision: 0.7418
Recall:    0.7465
F1-Score:  0.7308
CV Accuracy: 0.7254 (+/- 0.0135)
```

Figure : Model Evaluation Results

3.3.2 Confusion Matrix

Confusion matrices were used to analyze the classification performance by comparing actual and predicted labels. This visualization helps identify true positives, true negatives, false positives, and false negatives, which is particularly important in medical diagnosis tasks.

```
379     # Confusion matrices for top 3 models
380     top_3_models = sorted(results.items(), key=lambda x: x[1]['accuracy'], reverse=True)[:3]
381
382     fig, axes = plt.subplots(1, 3, figsize=(18, 5))
383
384     for idx, (name, result) in enumerate(top_3_models):
385         cm = confusion_matrix(y_test, result['y_pred'])
386         sns.heatmap(cm, annot=True, fmt='d', cmap='Blues', ax=axes[idx], cbar=False)
387         axes[idx].set_title(f'{name}\nAccuracy: {result["accuracy"]:.4f}', fontsize=12, fontweight='bold')
388         axes[idx].set_ylabel('True Label', fontsize=11)
389         axes[idx].set_xlabel('Predicted Label', fontsize=11)
390
391     plt.tight_layout()
392     plt.show()
```

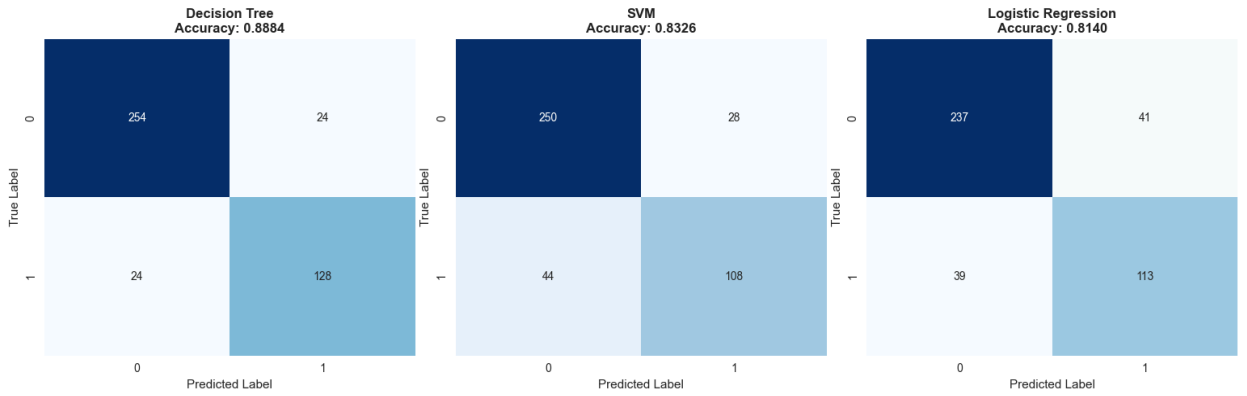

Figure 5.2: Confusion matrix of the Support Vector Machine (SVM) Top 3 models.

3.3.3 Model Performance Comparison

Bar charts were used to compare the performance of all classification models based on accuracy, precision, recall, and F1-score. This visualization provides a clear comparison of model effectiveness and highlights the best-performing model.

```
459     # Visualize results
460     print("\n[5] Visualizing Results...")
461     visualize_results(results, y_test)
```

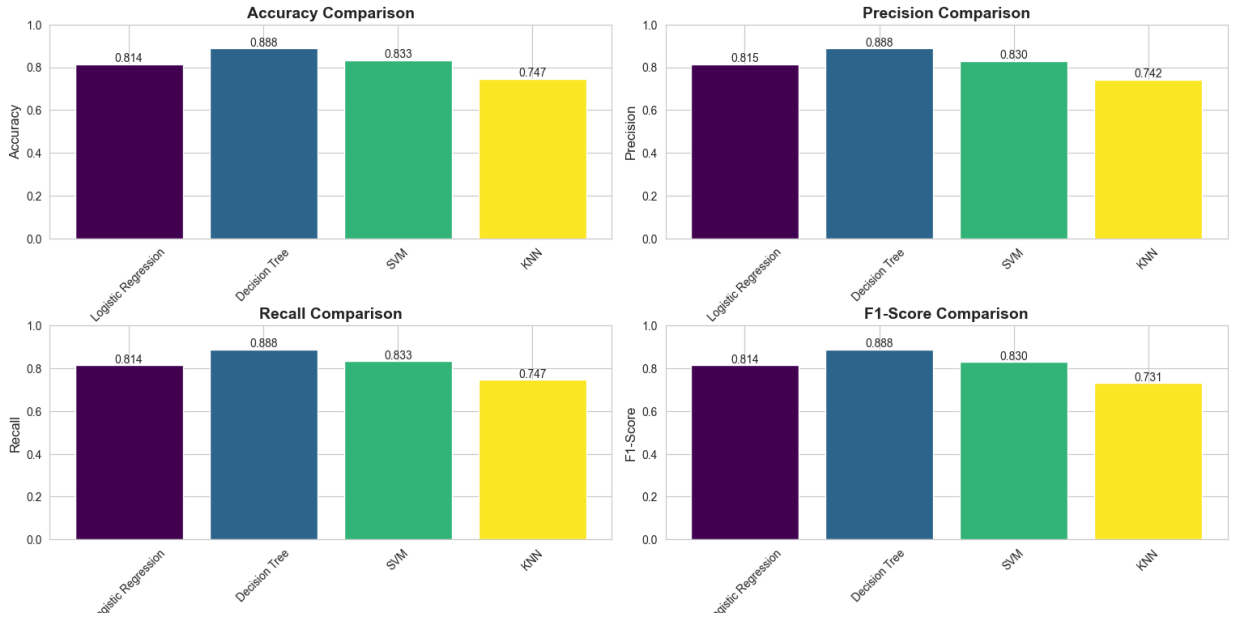

Figure : Performance comparison of classification models using multiple evaluation metrics.

3.3.4 ROC Curve Analysis

Receiver Operating Characteristic (ROC) curves were plotted to evaluate the trade-off between the true positive rate and false positive rate for each classification model. The Area Under the Curve (AUC) was used to assess the discriminative ability of the models.

```
    # ROC curves for models with probability predictions
    plt.figure(figsize=(10, 8))

    for name, result in results.items():
        if result['y_pred_proba'] is not None:
            fpr, tpr, _ = roc_curve(y_test, result['y_pred_proba'])
            auc = roc_auc_score(y_test, result['y_pred_proba'])
            plt.plot(fpr, tpr, label=f'{name} (AUC = {auc:.3f})', linewidth=2)

    plt.plot([0, 1], [0, 1], 'k--', label='Random Classifier', linewidth=2)
    plt.xlabel('False Positive Rate', fontsize=12)
    plt.ylabel('True Positive Rate', fontsize=12)
    plt.title('ROC Curves Comparison', fontsize=14, fontweight='bold')
    plt.legend(loc='lower right', fontsize=10)
    plt.grid(alpha=0.3)
    plt.tight_layout()
    plt.show()
```

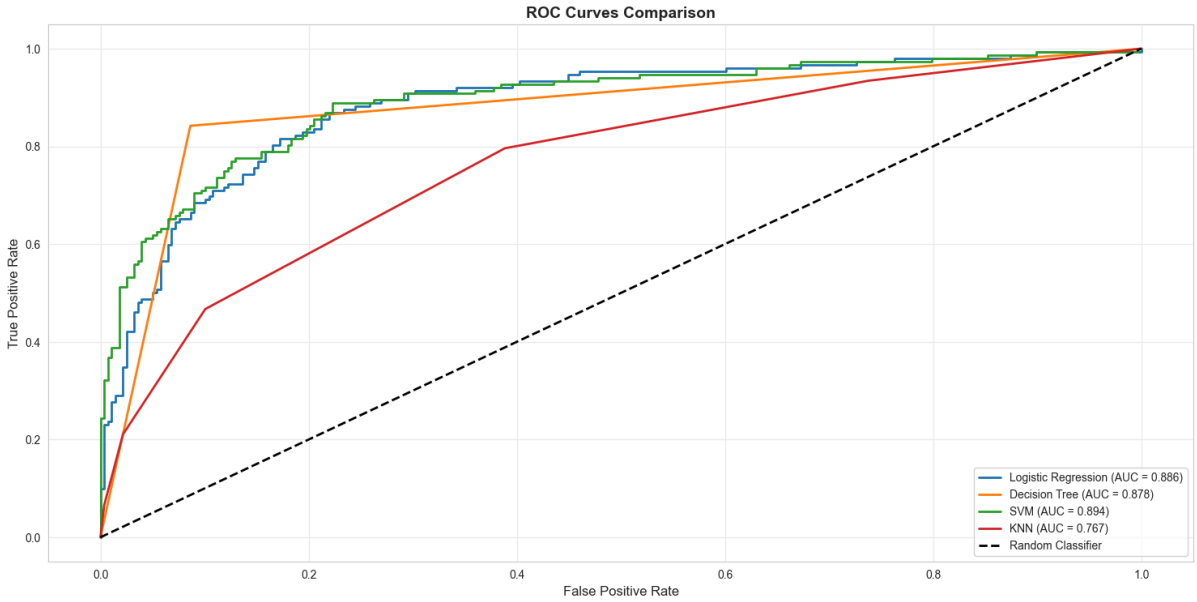

Figure : ROC curve comparison of classification models.

3.3.5 Prediction and Decision Making

All four models were successfully trained on the preprocessed dataset. The console output above shows the initial performance metrics obtained from the test set.

Model	Accuracy	Precision	Recall	F1- Score	CV Accuracy	ROC-AUC
Logistic Regression	0.8140	0.8145	0.8140	0.8142	0.8395	0.886
Decision Tree	0.8884	0.8884	0.8884	0.8884	0.9069	0.878
SVM	0.8326	0.8305	0.8326	0.8302	0.8336	0.894
KNN	0.7465	0.7418	0.7465	0.7308	0.7254	0.767

After model evaluation, the trained classification models were used to predict Alzheimer's disease status. Based on the predicted class and probability, a decision can be made to identify patients who are at higher risk of Alzheimer's disease, supporting early diagnosis and clinical decision-making.

```
304         # Make predictions
305         y_pred = model.predict(X_test)
306         y_pred_proba = model.predict_proba(X_test)[:, 1] if hasattr(model, 'predict_proba') else None
```

```
================================================================================
FINAL RESULTS SUMMARY
================================================================================

Best Model: Decision Tree
Accuracy: 0.8884
Precision: 0.8884
Recall: 0.8884
F1-Score: 0.8884
```

Figure : Example of Alzheimer's disease prediction using the best model.